Predicting Employee Absenteeism

Sarang Madhukar Kapse

7 December 2018

Contents

Chapter 1

Introduction

1.1 Problem Statement

XYZ is a courier company which is facing the problem of employee Absenteeism. As the primary operations of Courier Company such as collection, transportation and delivery mainly depend upon the human capital. The company is in loss. To minimize this loss the company has shared its data set and wants to know the solution for the following question asked by them:

Question 1:

1. What changes company should bring to reduce the number of absenteeism?

Question 2:

2. How much loss every month can we project in 2011 if same trend of absenteeism continues?

The aim of this project would be to answer the two questions, the solution for first question would be to check the variable importance of independent variables with the dependent variable. The solution for the second question would be to build a suitable model which would fit the dataset and predict the test cases accurately and then calculating the losses in hours of absenteeism as the most important thing is the human capital for the company.

1.2 Dataset

The data set shared by the company contains 21 variables and 740 observations.

Dataset Details:

Number of Variables: 21

Variable Information:

1. Individual identification (ID)
2. Reason for absence (ICD).

Absences attested by the International Code of Diseases (ICD) stratified into 21 Categories (I to XXI) as follows:

I Certain infectious and parasitic diseases
II Neoplasm
III Diseases of the blood and blood-forming organs and certain disorders involving the Immune mechanism
IV Endocrine, nutritional and metabolic diseases
V Mental and behavioral disorders
VI Diseases of the nervous system
VII Diseases of the eye and adnexa
VIII Diseases of the ear and mastoid process
IX Diseases of the circulatory system
X Diseases of the respiratory system
XI Diseases of the digestive system
XII Diseases of the skin and subcutaneous tissue
XIII Diseases of the musculoskeletal system and connective tissue
XIV Diseases of the genitourinary system
XV Pregnancy, childbirth and the puerperium
XVI Certain conditions originating in the perinatal period
XVII Congenital malformations, deformations and chromosomal abnormalities
XVIII Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
XIX Injury, poisoning and certain other consequences of external causes
XX External causes of morbidity and mortality
XXI Factors influencing health status and contact with health services.

And 7 categories without (CID) patient follow-up (22), medical consultation (23), blood donation (24), laboratory examination (25), unjustified absence (26), physiotherapy (27), dental consultation (28).

3. Month of absence
4. Day of the week (Monday (2), Tuesday (3), Wednesday (4), Thursday (5), Friday (6))
5. Seasons (summer (1), autumn (2), winter (3), spring (4))
6. Transportation expense
7. Distance from Residence to Work (kilometers)
8. Service time
9. Age

10. Work load Average/day
11. Hit target
12. Disciplinary failure (yes=1; no=0)
13. Education (high school (1), graduate (2), postgraduate (3), master and doctor (4))
14. Son (number of children)
15. Social drinker (yes=1; no=0)
16. Social smoker (yes=1; no=0)
17. Pet (number of pet)
18. Weight
19. Height
20. Body mass index
21. Absenteeism time in hours (target)

After data preprocessing and calculating the importance the number of variable reduces to 13 out of which 12 are independent and 1 is our target variable the following table shows the dataset for predicting the losses.

Independent Variables	**Dependent Variable**
Reason for absence	**Absenteeism time in hours**
Month of absence	
Seasons	
Transportation expense	
Distance from Residence to Work	
Service time	
Age	
Hit target	
Disciplinary failure	
Son	
Height	
Body mass index	

Table 1.2.1 Variables under predictions

Chapter 2

Methodology

2.1 Pre Processing

Any dataset which we want to use for building model is explored and necessary steps are taken after exploring the dataset which involves cleaning the missing values, detecting the outliers, normalizing the data observation and deleting variables which are multi-collinear and running the chi square and regression test to detect and delete variables which don't help us to predict our target variable. The preprocessing steps used on the dataset are shown by the following table.

Steps	Pre processing technique
Step 1	Changing the required data types
Step 2	Detect missing values and impute with KNN
Step 3	Box plot distribution and outlier check
Step 4	Detect outliers and impute with KNN
Step 5	Feature selection with Co-relation plot and Chi square test .
Step 6	Normality check and normalizing the dataset observation

Table 2.1 Preprocessing steps

Step 1: Changing the require Data types

After loading our data set and running the str(data_absent) command in R we can see that most of the variables which are actually factorial is named as numeric and vice-versa. In order to perform KNN imputation on our missing values we need to change the data types as the KNN imputation method works on the concept of replacing values according to the data types.

```
> str(data_absent) # Checking the required data types
'data.frame':    740 obs. of  21 variables:
 $ ID                              : int  11 36 3 7 11 3 10 20 14 1 ...
 $ Reason.for.absence              : int  26 0 23 7 23 23 22 23 19 22 ...
 $ Month.of.absence                : int  7 7 7 7 7 7 7 7 7 7 ...
 $ Day.of.the.week                 : int  3 3 4 5 5 6 6 6 2 2 ...
 $ Seasons                         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Transportation.expense          : int  289 118 179 279 289 179 NA 260 155 235 ...
 $ Distance.from.Residence.to.Work : int  36 13 51 5 36 51 52 50 12 11 ...
 $ Service.time                    : int  13 18 18 14 13 18 3 11 14 14 ...
 $ Age                             : int  33 50 38 39 33 38 28 36 34 37 ...
 $ Work.load.Average.day           : Factor w/ 38 levels "205,917","222,196"
 $ Hit.target                      : int  97 97 97 97 97 97 97 97 97 97 ...
 $ Disciplinary.failure            : int  0 1 0 0 0 0 0 0 0 0 ...
 $ Education                       : int  1 1 1 1 1 1 1 1 1 3 ...
 $ Son                             : int  2 1 0 2 2 0 1 4 2 1 ...
 $ Social.drinker                  : int  1 1 1 1 1 1 1 1 1 0 ...
 $ Social.smoker                   : int  0 0 0 1 0 0 0 0 0 0 ...
 $ Pet                             : int  1 0 0 0 1 0 4 0 0 1 ...
 $ Weight                          : int  90 98 89 68 90 89 80 65 95 88 ...
 $ Height                          : int  172 178 170 168 172 170 172 168 196 172 ...
 $ Body.mass.index                 : int  30 31 31 24 30 31 27 23 25 29 ...
 $ Absenteeism.time.in.hours       : int  4 0 2 4 2 NA 8 4 40 8 ...
```

The marked variables need to be changed as these are factorial variables but defined as integer variable in the dataset.

After using this command in R the required data types were changed.

```
data_absent$Reason.for.absence   = as.factor(data_absent$Reason.for.absence)
data_absent$Month.of.absence     = as.factor(data_absent$Month.of.absence)
data_absent$Day.of.the.week      = as.factor(data_absent$Day.of.the.week)
data_absent$Seasons              = as.factor(data_absent$Seasons)
data_absent$Disciplinary.failure = as.factor(data_absent$Disciplinary.failure)
data_absent$Education            = as.factor(data_absent$Education)
data_absent$Son                  = as.factor(data_absent$Son)
data_absent$Social.drinker       = as.factor(data_absent$Social.drinker)
data_absent$Social.smoker        = as.factor(data_absent$Social.smoker)
data_absent$Pet                  = as.factor(data_absent$Pet)
```

Step 2 : Detect missing values and impute with KNN

After changing the required data types, we proceed to our next step which is detecting the missing values and imputing it with KNN. We are using KNN method because our dataset has both categorical and numerical variables and KNN method can be used for both types of variables. The following histogram shows the missing value percentages of our dataset.

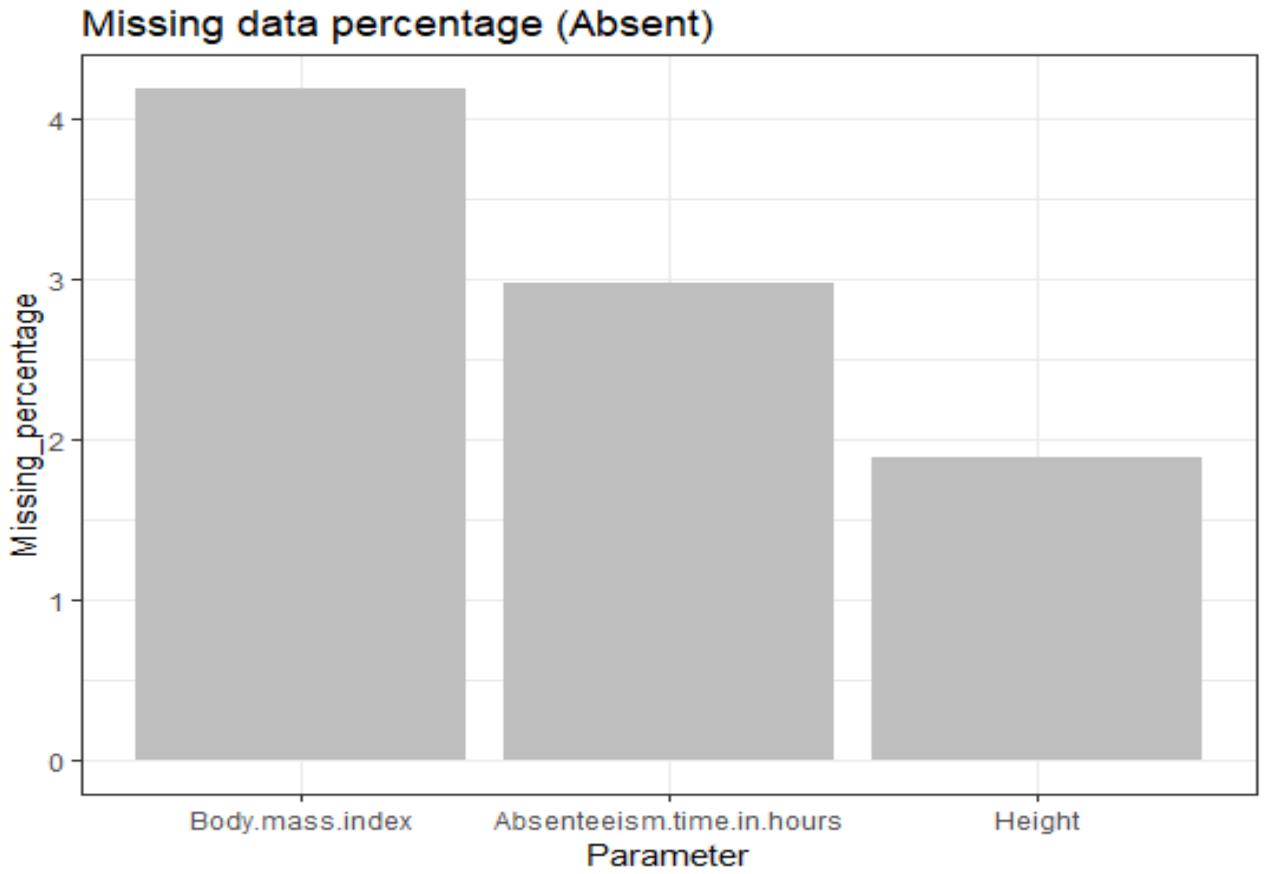


Figure 2.1.1 Missing value percentage

As we could see that there are parameters with missing values upto 4 percent we need to replace this missing values with related dataset values because while applying machine learning algorithm be it supervised or unsupervised random dataset observation are selected to generate our test and train dataset and each observation plays an important role for predicting our target variable. Another way could be deleting these missing observation but it should be implemented when the dataset contains missing values in a very large quantity for example in our dataset if the missing value percentage of Height would have been greater than 50 % then these observation would not have been replaced with KNN imputation because these observation cannot be considered as real observation.

Step 3: Box plot distribution and outlier check

A dataset containing outliers can greatly affect the model developement process. The outlier observations in our dataset is detected graphically using box plot method the following figure shows the box plot distribution for the continuous variables.

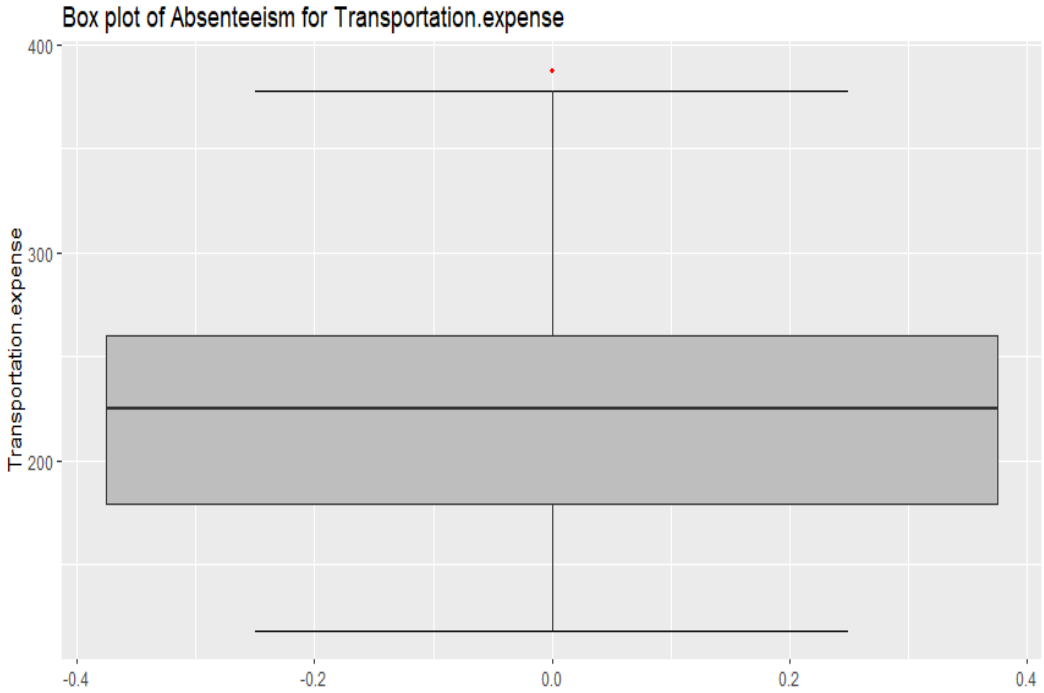


Figure 2.1.2 Box plot for Transportation Expense Variable

The above box plot shows a red dot which is above our upper fence in the figure this red dot is nothing but our outlier observation. Likewise the following figure shows the outliers present in various variable in our dataset.

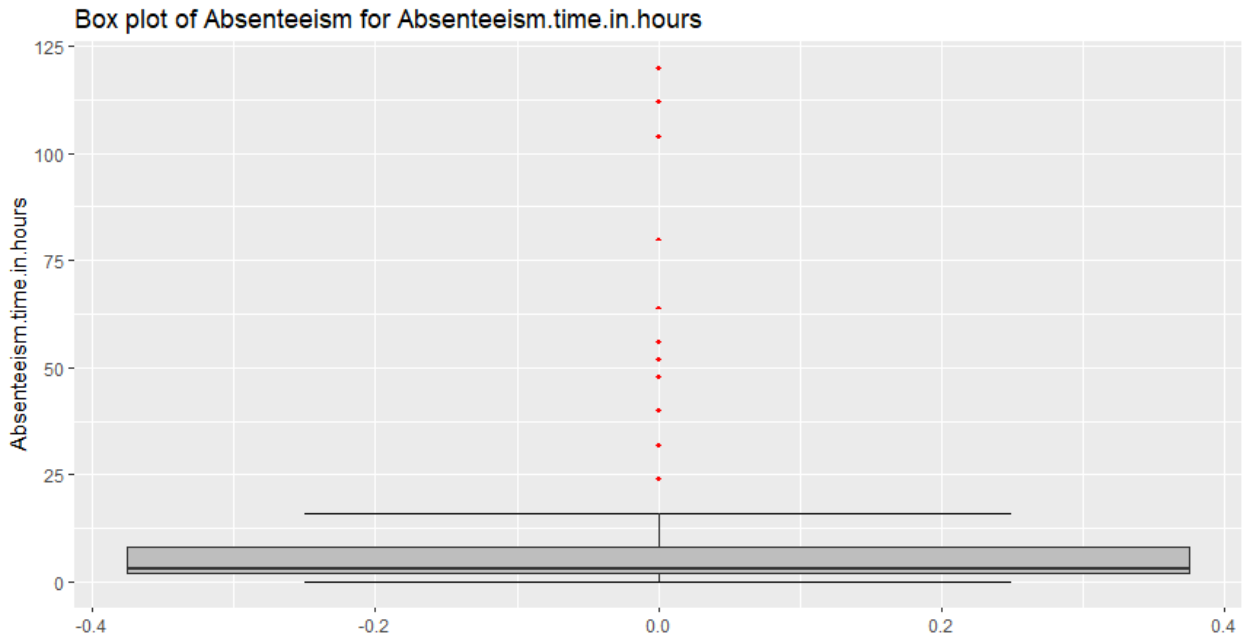


Figure 2.1.3 Box Plot for Absenteeism time in hours

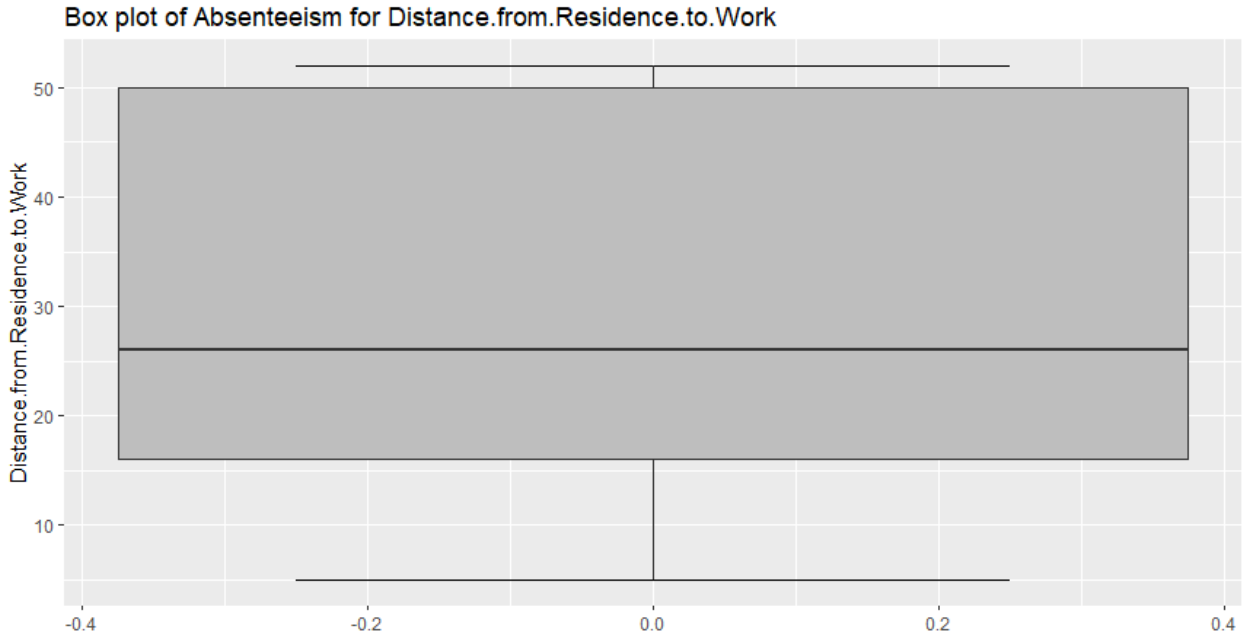


Figure 2.1.4 Box Plot for Distance from Residence to work

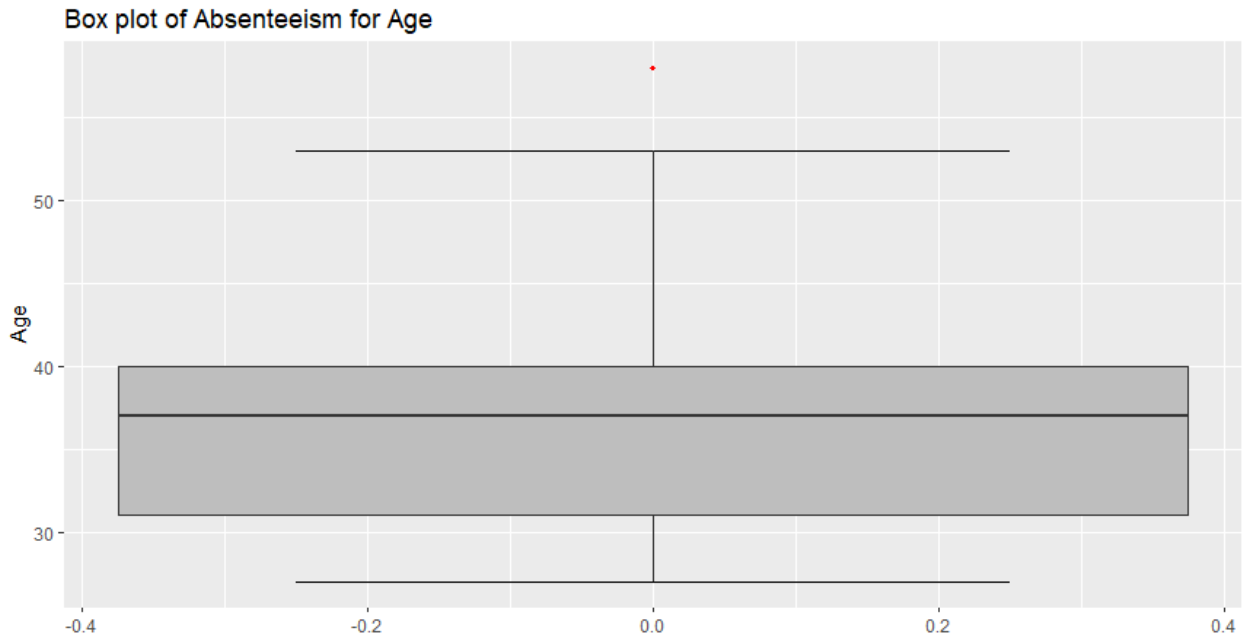


Figure 2.1.5 Box plot for Age Variable

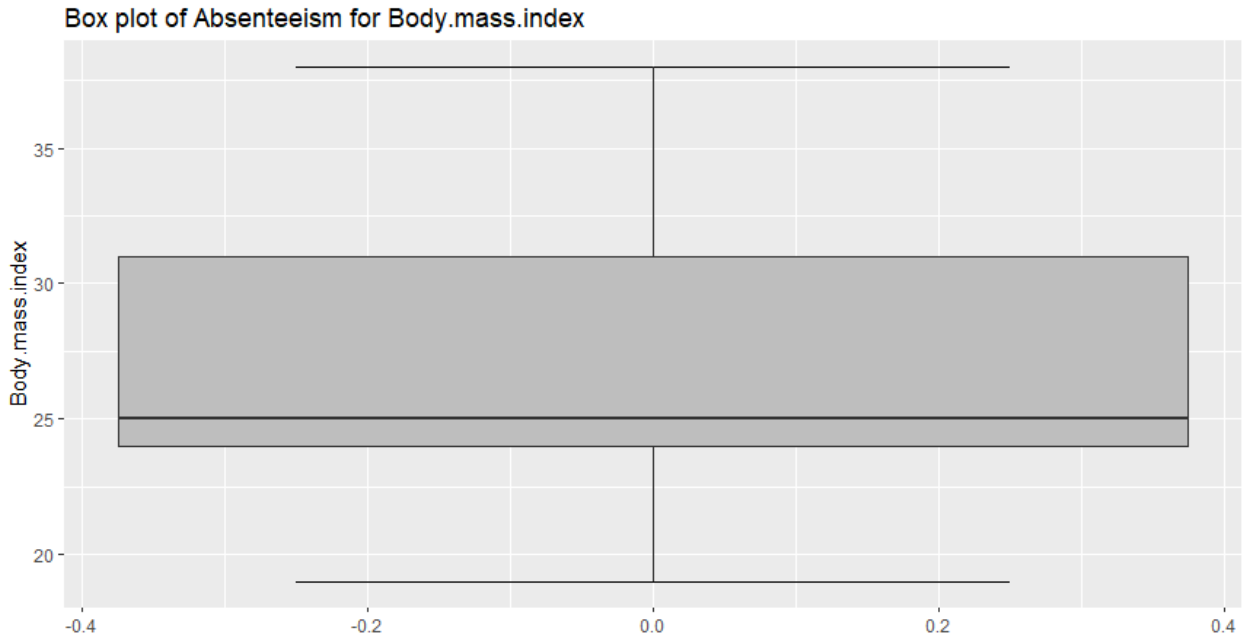


Figure 2.1.6 Box Plot for Body Mass Index

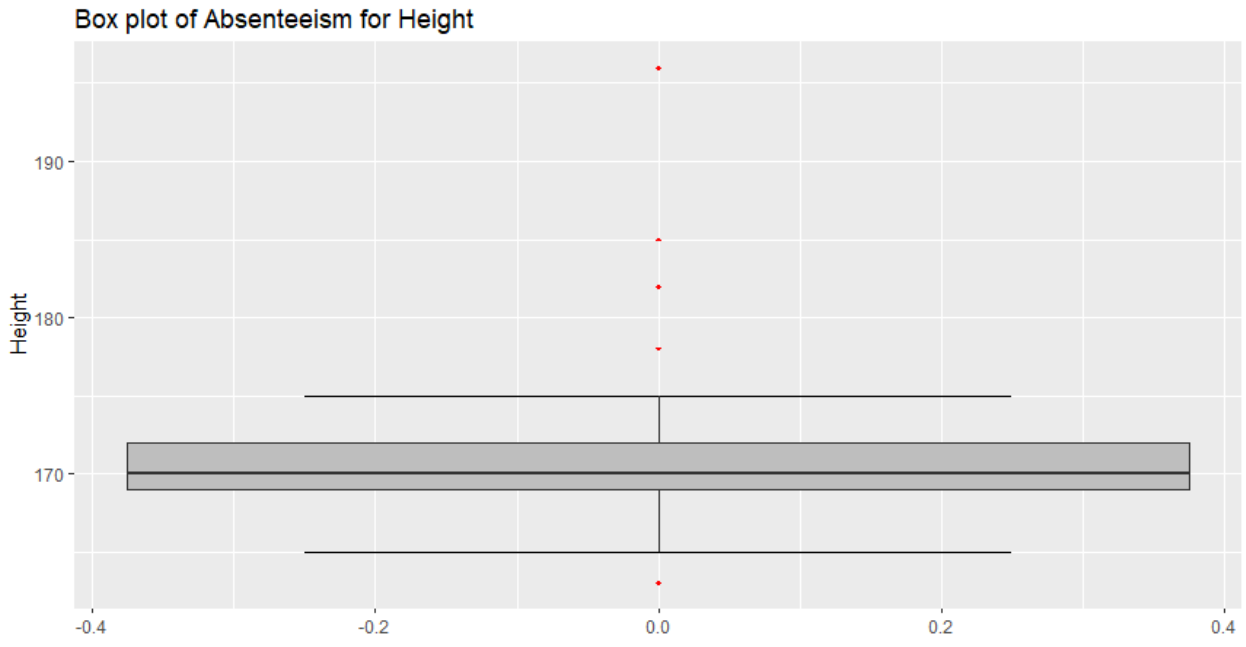


Figure 2.1.7 Box plot for Height

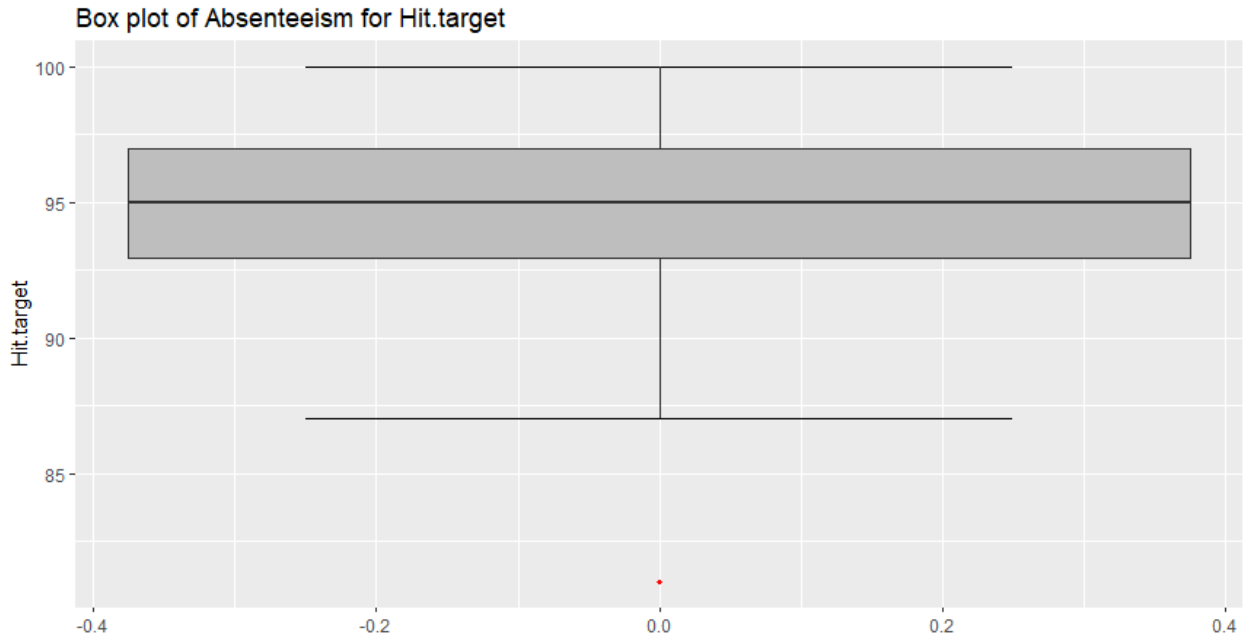


Figure 2.1.8 Box Plot for Hit Target

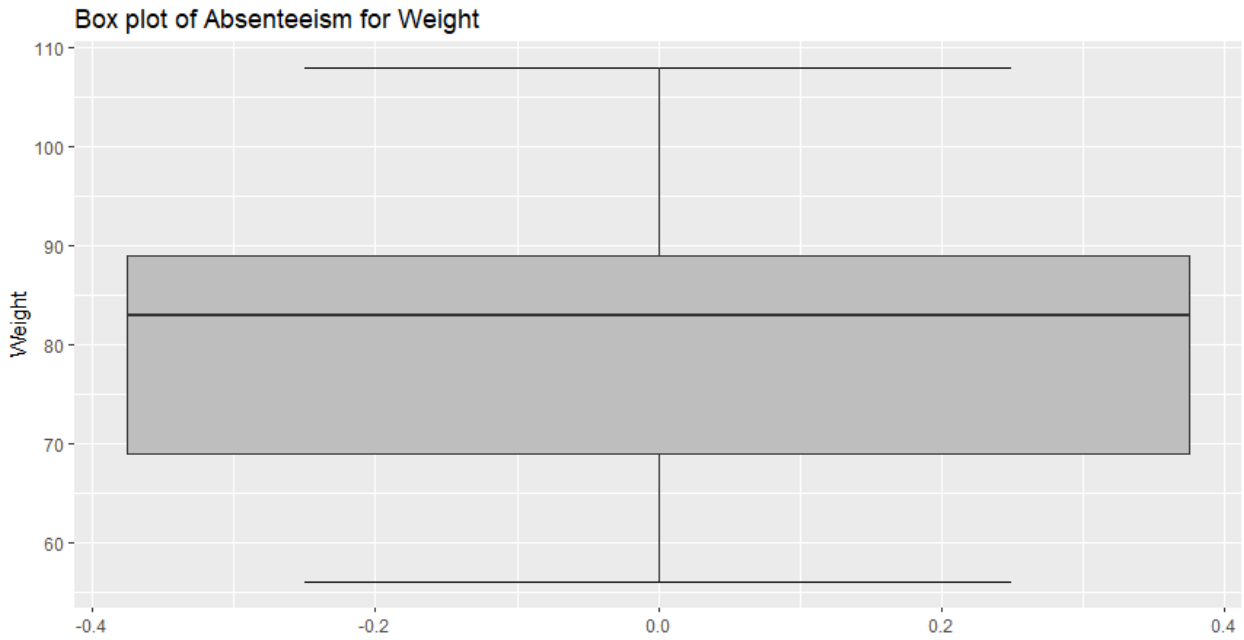


Figure 2.1.9 Box Plot for Weight

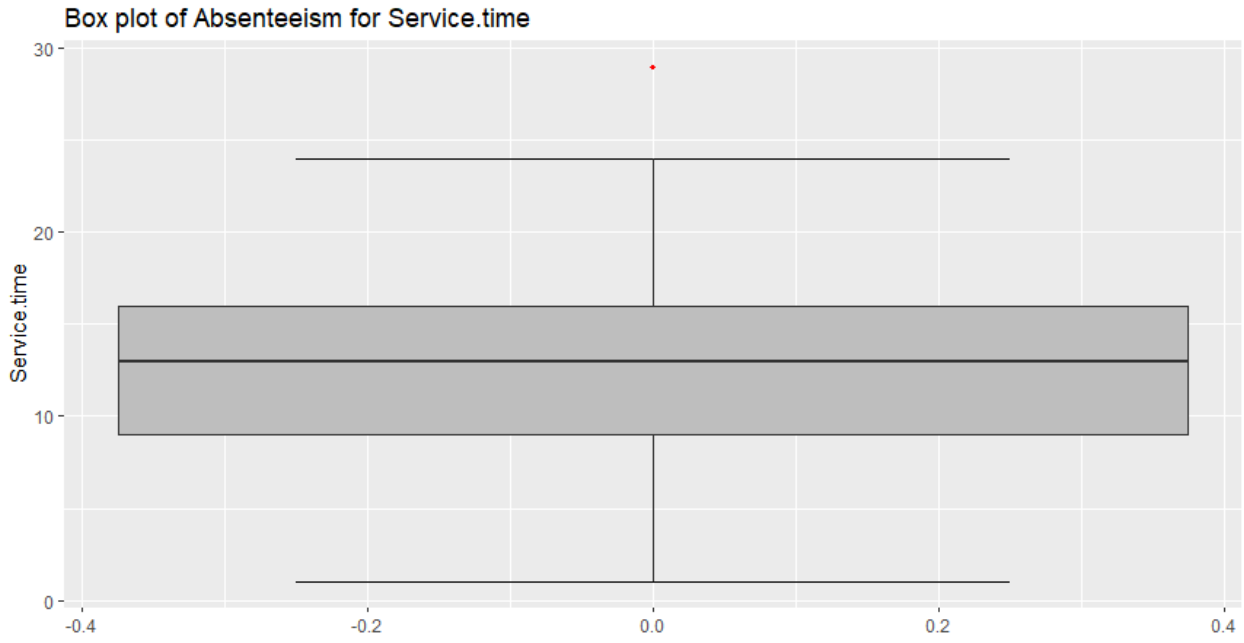


Figure 2.1.10 Box Plot for Service time

The above box plot detection method shows that there are some points in the variables of our dataset which need to be removed or impute as they lie above the upper fence. As the deletion process would delete the entire rows containing outliers in our dataset the method that was used to treat this outliers was KNN imputation method that we used earlier during missing value analysis.

############### loop to remove outlier and impute using KNN#############

```
for(i in cnames_absent)
  {val = data_absent[,i][data_absent[,i] %in% boxplot.stats(data_absent[,i])$out]
  #print(length(val))
  data_absent[,i][data_absent[,i] %in% val] = NA
  }

data_absent = knnImputation(data_absent, k = 7)
```

Step 5 : Feature selection with Correlation plot and Chi square test .

In order start our model development process we also need to see multi-co linearity issues in our dataset for that we need to see the correlation plot and check the variables which are multicollinear. Multi collinear variables are those variables which carry the same information and don't add any extra information but the same information as their partner variables we need to detect these variables and remove them as during our model building and predicting the test data these consume unnecessary space and affect our predictions. The following figure shows the correlation plot.

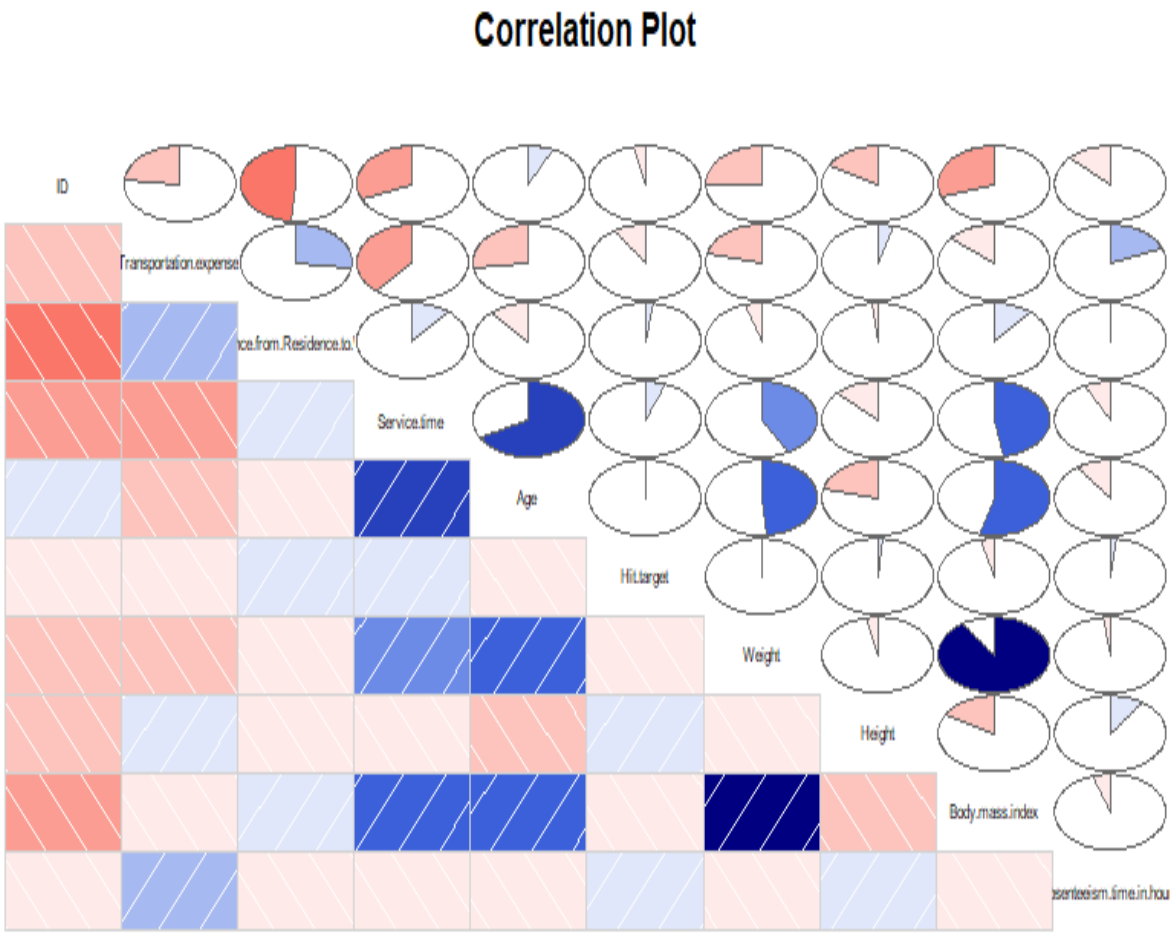


Figure 2.1.11 Correlation Plot

From the correlation plot it can be seen that the two variables Body Mass Index and Weight are having high multicollinear relationship as the circle is dark blue. So we can remove one of the variables from our data set. Note that the correlation plot is only used for continuous variable.

Chi square Test of Independence :
The correlation plot is used to check the relationship between two continuous variable but what about the variables which are factors in such cases we need to use the chi square test of Independence. The chi square test is based on two assumptions Null and Alternate hypothesis after running this test we need to select one of our assumptions. In our case we are selecting Absenteeism time in hours as our target variable and the rest of the variables as our Independent or predictor variable.

Two assumptions:

1. Null Hypothesis:

Target variable and the Predictor variables are independent.

2. Alternate Hypothesis:

Target Variable and the Predictor variables are not independent i.e. they are dependent.

After running the test we get the results in form of X-squared , df , p-value . From these values we need to see the parameter p-value . If the p-value is less than 0.05 then we need to reject null hypothesis. In short if P value is greater than 0.05 then we need to drop that variable from our data set.

The results of chi square test are as follows:

[1] "Reason.for.absence"

Pearson's Chi-squared test

data: table(data_absent$Absenteeism.time.in.hours, factor_data[, i])

X-squared = 3496.2, df = 1971, p-value < 2.2e-16

[2] "Month.of.absence"

Pearson's Chi-squared test

data: table(data_absent$Absenteeism.time.in.hours, factor_data[, i])

X-squared = 971.21, df = 876, p-value = 0.01346

[3] "Day.of.the.week"

Pearson's Chi-squared test

data: table(data_absent$Absenteeism.time.in.hours, factor_data[, i])

X-squared = 289.12, df = 292, p-value = 0.5366

[4] "Seasons"

Pearson's Chi-squared test

data: table(data_absent$Absenteeism.time.in.hours, factor_data[, i])

X-squared = 282.66, df = 219, p-value = 0.002402

[5] "Work.load.Average.day"

Pearson's Chi-squared test

data: table(data_absent$Absenteeism.time.in.hours, factor_data[, i])

X-squared = 2956.5, df = 2701, p-value = 0.0003601

[6] "Disciplinary.failure"

Pearson's Chi-squared test

data: table(data_absent$Absenteeism.time.in.hours, factor_data[, i])

X-squared = 668.78, df = 73, p-value < 2.2e-16

[7] "Education"

Pearson's Chi-squared test

data: table(data_absent$Absenteeism.time.in.hours, factor_data[, i])

X-squared = 112.01, df = 219, p-value = 1

[8] "Son"

Pearson's Chi-squared test

data: table(data_absent$Absenteeism.time.in.hours, factor_data[, i])

X-squared = 477.88, df = 292, p-value = 3.755e-11

[9] "Social.drinker"

Pearson's Chi-squared test

data: table(data_absent$Absenteeism.time.in.hours, factor_data[, i])

X-squared = 81.702, df = 73, p-value = 0.2272

[10] "Social.smoker"

Pearson's Chi-squared test

data: table(data_absent$Absenteeism.time.in.hours, factor_data[, i])

X-squared = 90.105, df = 73, p-value = 0.08499

[11] "Pet"

Pearson's Chi-squared test

data: table(data_absent$Absenteeism.time.in.hours, factor_data[, i])

X-squared = 293.19, df = 365, p-value = 0.9977

From the correlation plot and chi square test we can select variables which are necessary for model selection and development and reduce the dimension of our dataset. The following command is used for dimension reduction.

```
################### Dimension Reduction################
data_absent = subset(data_absent,
                     select = -
c(ID,Weight,Day.of.the.week,Education,Social.smoker,Social.drinker,Pet,Work.load.Average.day))
```

After the dimension reduction we get the dataset with 13 variables and 740 observation.

Following are the variables

1 .Reason for absence

2.Month of absence

3.Seasons

4.Transportation expense

5. Distance from Residence to Work

6. Service time

7.Age

8.Hit target

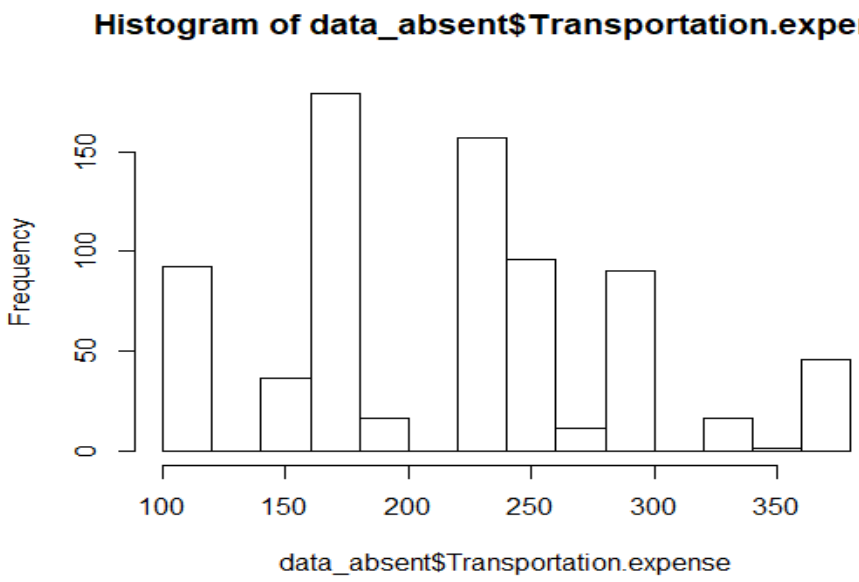
9. Disciplinary failure

10. Son

11. Height

12.Bodymass index

13. Absenteeism time in hours



st preprocessing step is
vation are in thousands

while the Body mass
of model development
first check whether our
hen we would go for
ıe data first.

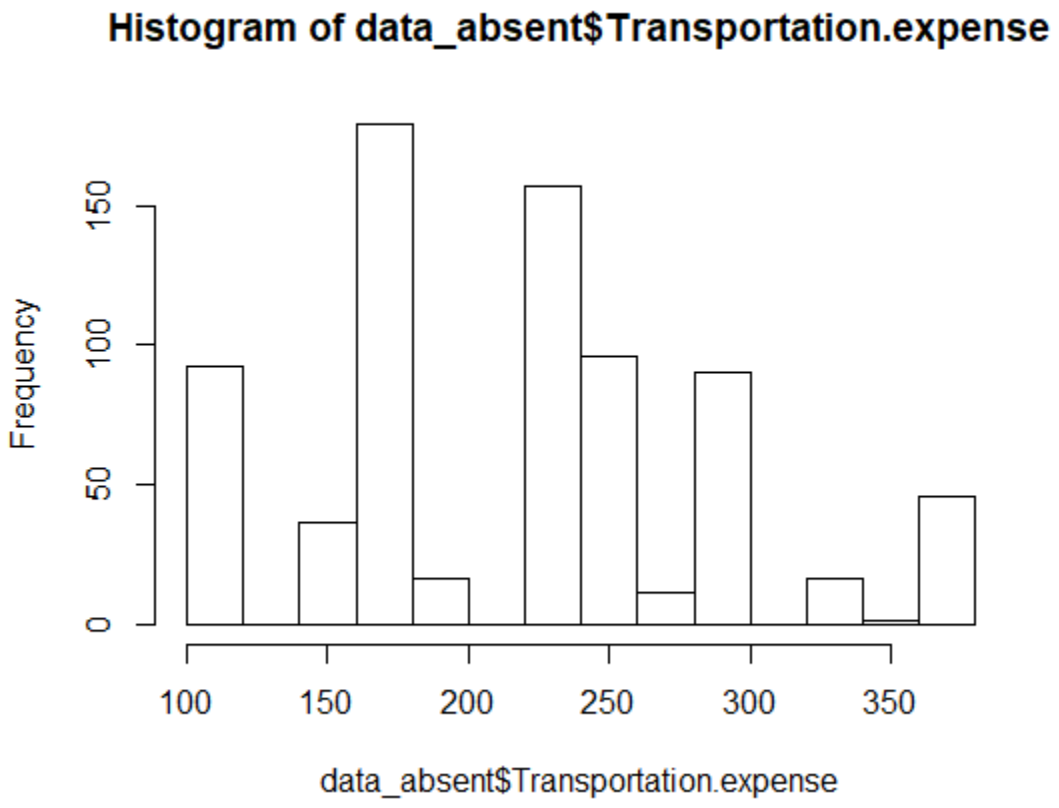


Figure 2.1.12 Histogram for Transportation expenses

The above histogram for transportation expenses shows that the data is right skewed and needs to be normalized.

For normalizing the dataset we used the following R code

```
cnames_norm = c("Transportation.expense","Distance.from.Residence.to.Work","Service.time","Age","Hit.target","Height","Body.mass.index","Absenteeism.time.in.hours")

for(i in cnames_norm){
  print(i)
  data_absent[,i] = (data_absent[,i] - min(data_absent[,i]))/
    (max(data_absent[,i] - min(data_absent[,i])))
}
```

The range of normalized data observation is from 0 to 1.

2.2 Modeling

2.2.1 Model Selection

During our preprocessing steps we realized that our dataset has both Categorical and Continuous variable . But our target variable Absenteeism time in hours is continuous variable and we need to predict this variable after selecting our model. In our case Decision tree for regression as target variable and Linear regression model was used to answer the two questions asked by the company. From the two models we will compare the predictions and select the better one between these models.

2.2.2 Decision tree for regression

Decision tree is a very simple model which is based on a branching series of Boolean test. It can be used for both classification and regression
In our case the target variable Absenteeism time in hours is continuous variable so we would use decision tree for regression.

Let us see how the decision tree is used for predicting the test cases.

After running the following code of chunk in R ,

```
#######################Decision tree  for regression ####################

fit_dt = rpart(Absenteeism.time.in.hours~ ., data = df, method = "anova")

#Predict for new test cases
predictions_DT = predict(fit_dt, test[,-13])

RMSE(predictions_DT, test$Absenteeism.time.in.hours
```

Result is,

```
> RMSE(predictions_DT, test$Absenteeism.time.in.hours)
[1] 0.1661673
```

The RMSE which is root mean square error gives value of 0.16616 which is very close to 0 . The closer the value of RMSE to zero the better the predicted values are from actual values.

2.2.3 Linear Regression

Before going for the prediction of test cases in regression model, first we would check the variance inflation factor and the multi-co linearity factor for our dataset the VIF . The following result help us to know whether our dataset is suitable for model building.

vifcor(cnames_df_numeric[, -8], th = 0.9)
No variable from the 7 input variables has collinearity problem.

The linear correlation coefficients ranges between:
min correlation (Hit.target ~ Age): -0.003713532
max correlation (Age ~ Service.time): 0.6682232

---------- VIFs of the remained variables --------

	Variables	VIF
1	Transportation.expense	1.357189
2	Distance.from.Residence.to.Work	1.275779
3	Service.time	2.298234
4	Age	2.310935
5	Hit.target	1.017519
6	Height	1.061569
7	Body.mass.index	1.515783

From the test it is clear that No variable from the 7 input variables has collinearity problem. Note that cnames_df_numeric is a subset of our dataset df with 13 variables which include 8 numeric and 5 factorial variables.

Now let us move towards our model building process. The result of our linear regression model building on the train data is as follows

```
> lm_model = lm(Absenteeism.time.in.hours ~., data = train)
>
> #Summary of the model
> summary(lm_model)

Call:
lm(formula = Absenteeism.time.in.hours ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.38793 -0.08194 -0.00724  0.05911  0.79780

Coefficients: (1 not defined because of singularities)
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          -0.06226    0.18273  -0.341 0.733463
Reason.for.absence1   0.44088    0.05345   8.248 1.23e-15 ***
Reason.for.absence2   0.08922    0.16613   0.537 0.591447
Reason.for.absence3   0.53940    0.16701   3.230 0.001315 **
Reason.for.absence4   0.33708    0.09959   3.385 0.000764 ***
Reason.for.absence5   0.40821    0.09993   4.085 5.08e-05 ***
Reason.for.absence6   0.32157    0.07291   4.410 1.25e-05 ***
Reason.for.absence7   0.26208    0.05618   4.665 3.90e-06 ***
Reason.for.absence8   0.23849    0.10045   2.374 0.017934 *
Reason.for.absence9   0.55324    0.09853   5.615 3.15e-08 ***
Reason.for.absence10  0.38076    0.04704   8.095 3.83e-15 ***
Reason.for.absence11  0.30703    0.04985   6.158 1.43e-09 ***
Reason.for.absence12  0.25557    0.07385   3.461 0.000581 ***
Reason.for.absence13  0.34045    0.03837   8.874  < 2e-16 ***
Reason.for.absence14  0.24709    0.05047   4.895 1.30e-06 ***
Reason.for.absence15  0.45137    0.16697   2.703 0.007080 **
Reason.for.absence16  0.06767    0.09967   0.679 0.497444
Reason.for.absence17  0.41815    0.16625   2.515 0.012187 *
Reason.for.absence18  0.38447    0.05138   7.483 2.97e-13 ***
Reason.for.absence19  0.41024    0.04426   9.270  < 2e-16 ***
Reason.for.absence21  0.40261    0.07998   5.034 6.56e-07 ***
Reason.for.absence22  0.37690    0.04446   8.476 2.22e-16 ***
Reason.for.absence23  0.13709    0.03427   4.000 7.21e-05 ***
Reason.for.absence24  0.39240    0.12015   3.266 0.001161 **
Reason.for.absence25  0.18357    0.04647   3.950 8.85e-05 ***
Reason.for.absence26  0.37862    0.04373   8.659  < 2e-16 ***
Reason.for.absence27  0.12027    0.04301   2.796 0.005350 **
Reason.for.absence28  0.13747    0.03514   3.912 0.000103 ***
Month.of.absence1    -0.04643    0.17709  -0.262 0.793285
Month.of.absence2     0.01054    0.17659   0.060 0.952424
Month.of.absence3     0.04987    0.17418   0.286 0.774762
Month.of.absence4    -0.03034    0.17237  -0.176 0.860338
Month.of.absence5    -0.05712    0.17175  -0.333 0.739560
```

```
Month.of.absence6               -0.02502    0.17207  -0.145 0.884440
Month.of.absence7                0.02862    0.17909   0.160 0.873074
Month.of.absence8                0.02178    0.17922   0.122 0.903303
Month.of.absence9                0.02249    0.17875   0.126 0.899904
Month.of.absence10               0.01126    0.18144   0.062 0.950538
Month.of.absence11              -0.01873    0.18156  -0.103 0.917871
Month.of.absence12               0.01209    0.18016   0.067 0.946500
Seasons2                         0.02398    0.05479   0.438 0.661842
Seasons3                         0.07850    0.04857   1.616 0.106610
Seasons4                         0.02532    0.04674   0.542 0.588216
Transportation.expense           0.11987    0.04267   2.809 0.005151 **
Distance.from.Residence.to.Work -0.07702    0.03247  -2.372 0.018042 *
Service.time                     0.06746    0.06816   0.990 0.322710
Age                             -0.06253    0.04905  -1.275 0.202916
Hit.target                      -0.01989    0.04134  -0.481 0.630574
Disciplinary.failure1                NA         NA      NA       NA
Son1                            -0.04678    0.01933  -2.420 0.015861 *
Son2                             0.01069    0.02334   0.458 0.647201
Son3                            -0.04122    0.06022  -0.684 0.494000
Son4                             0.07866    0.03799   2.071 0.038859 *
Height                           0.07692    0.04445   1.730 0.084143 .
Body.mass.index                  0.08767    0.04311   2.034 0.042490 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1602 on 540 degrees of freedom
Multiple R-squared:  0.4655,                    Adjusted R-squared:  0.413
F-statistic: 8.874 on 53 and 540  DF,                p-value: < 2.2e-16
```

The model development results such as the significance code and P value are important parameter to tell us about the model selected is going to work on our test data or not. The P value which is very less gives us assurance that the model selected is going to work very well on our test data.

Now, let us look at the variable importance and parameters calculated after the model is tested. The following code gives us the variable importance.

varImp(fit)*100

Variable Name	Importance in Percentage
Reason for absence	**116.013044**
Son	**50.627409**
Month of absence	**46.767400**
Transportation expense	**31.621362**
Body mass index	**27.439138**
Service time	**23.492444**
Age	**20.249707**
Disciplinary failure	**13.054967**
Distance from Residence to Work	**8.521611**
Height	**7.081810**
Hit target	**6.770837**
Seasons	**4.837035**

Table 2.2.1 Variable Importance

The above table shows the importance of variable to predict the test cases. From the above table it is clear that the most important variable for predicting our target variable Absenteeism time in hours is Reason of absence while the least important is the seasons.

From the linear regression model we get predicted values using the code

```
#Predict for new test cases
predictions_DT = predict(fit, test[,-13])
```

but these predicted values of test cases and actual value of the test cases needs to be evaluated. In order to compare them we have used the following command.

```
regr.eval(test[,'Absenteeism.time.in.hours'] , predictions_DT ,stats = c('mae','rmse','mape','mse'))
```

with this command in R we get the following results.

mae	rmse	mape	mse
0.1179884	0.1835296	Inf	0.0336831

The above results with mean absolute error, root mean square error, mean absolute percentage error and mean square error. The important parameter is root mean square error which gives us the indication of how far our predicted values are away from the actual test values. The result shows that it is 0.1835, the lesser it is the better is our model for predicting the values. In our case considering 0 as the perfect value for our prediction and 1 as the worst case our model is (100 – 18.35 = 81.65 %) accurately predicting the values.

Chapter 3

Conclusion

3.1 Model Evaluation

Now that we have used decision tree for regression and Linear regtression model, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive Performance
2. Interpretability
3. Computational Efficiency

In our case Predictive performance and Computational efficiency plays an important role for answering the two questions put forth by the company. The answer for the first question which to give advice to the company for making such changes which would help them to reduce the employee absenteeism is found in our regression model analysis where the significance code during model building will be helpful to know which variable performance should be increase or decrease to reduce absenteeism rate.

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	**-0.06226**	**0.18273**	**-0.341**	**0.733463**	
Reason.for.absence1	**0.44088**	**0.05345**	**8.248**	**1.23e-15**	*******
Reason.for.absence2	**0.08922**	**0.16613**	**0.537**	**0.591447**	
Reason.for.absence3	**0.53940**	**0.16701**	**3.230**	**0.001315**	******
Reason.for.absence4	**0.33708**	**0.09959**	**3.385**	**0.000764**	*******
Reason.for.absence5	**0.40821**	**0.09993**	**4.085**	**5.08e-05**	*******
Reason.for.absence6	**0.32157**	**0.07291**	**4.410**	**1.25e-05**	*******
Reason.for.absence7	**0.26208**	**0.05618**	**4.665**	**3.90e-06**	*******
Reason.for.absence8	**0.23849**	**0.10045**	**2.374**	**0.017934**	*****
Reason.for.absence9	**0.55324**	**0.09853**	**5.615**	**3.15e-08**	*******
Reason.for.absence10	**0.38076**	**0.04704**	**8.095**	**3.83e-15**	*******
Reason.for.absence11	**0.30703**	**0.04985**	**6.158**	**1.43e-09**	*******
Reason.for.absence12	**0.25557**	**0.07385**	**3.461**	**0.000581**	*******
Reason.for.absence13	**0.34045**	**0.03837**	**8.874**	**< 2e-16**	*******
Reason.for.absence14	**0.24709**	**0.05047**	**4.895**	**1.30e-06**	*******
Reason.for.absence15	**0.45137**	**0.16697**	**2.703**	**0.007080**	******
Reason.for.absence16	**0.06767**	**0.09967**	**0.679**	**0.497444**	
Reason.for.absence17	**0.41815**	**0.16625**	**2.515**	**0.012187**	*****
Reason.for.absence18	**0.38447**	**0.05138**	**7.483**	**2.97e-13**	*******
Reason.for.absence19	**0.41024**	**0.04426**	**9.270**	**< 2e-16**	*******
Reason.for.absence21	**0.40261**	**0.07998**	**5.034**	**6.56e-07**	*******

Reason.for.absence22	**0.37690**	**0.04446**	**8.476**	**2.22e-16 *** **
Reason.for.absence23	**0.13709**	**0.03427**	**4.000**	**7.21e-05 *** **
Reason.for.absence24	**0.39240**	**0.12015**	**3.266**	**0.001161 ** **
Reason.for.absence25	**0.18357**	**0.04647**	**3.950**	**8.85e-05 *** **
Reason.for.absence26	**0.37862**	**0.04373**	**8.659**	**< 2e-16 *** **
Reason.for.absence27	**0.12027**	**0.04301**	**2.796**	**0.005350 ** **
Reason.for.absence28	**0.13747**	**0.03514**	**3.912**	**0.000103 *** **
Month.of.absence1	**-0.04643**	**0.17709**	**-0.262**	**0.793285**
Month.of.absence2	**0.01054**	**0.17659**	**0.060**	**0.952424**
Month.of.absence3	**0.04987**	**0.17418**	**0.286**	**0.774762**
Month.of.absence4	**-0.03034**	**0.17237**	**-0.176**	**0.860338**
Month.of.absence5	**-0.05712**	**0.17175**	**-0.333**	**0.739560**
Month.of.absence6	**-0.02502**	**0.17207**	**-0.145**	**0.884440**
Month.of.absence7	**0.02862**	**0.17909**	**0.160**	**0.873074**
Month.of.absence8	**0.02178**	**0.17922**	**0.122**	**0.903303**
Month.of.absence9	**0.02249**	**0.17875**	**0.126**	**0.899904**
Month.of.absence10	**0.01126**	**0.18144**	**0.062**	**0.950538**
Month.of.absence11	**-0.01873**	**0.18156**	**-0.103**	**0.917871**
Month.of.absence12	**0.01209**	**0.18016**	**0.067**	**0.946500**
Seasons2	**0.02398**	**0.05479**	**0.438**	**0.661842**
Seasons3	**0.07850**	**0.04857**	**1.616**	**0.106610**
Seasons4	**0.02532**	**0.04674**	**0.542**	**0.588216**
Transportation.expense	**0.11987**	**0.04267**	**2.809**	**0.005151 ** **
Distance.from.Residence.to.Work	**-0.07702**	**0.03247**	**-2.372**	**0.018042 * **
Service.time	**0.06746**	**0.06816**	**0.990**	**0.322710**
Age	**-0.06253**	**0.04905**	**-1.275**	**0.202916**
Hit.target	**-0.01989**	**0.04134**	**-0.481**	**0.630574**
Disciplinary.failure1	**NA**	**NA**	**NA**	**NA**
Son1	**-0.04678**	**0.01933**	**-2.420**	**0.015861 * **
Son2	**0.01069**	**0.02334**	**0.458**	**0.647201**
Son3	**-0.04122**	**0.06022**	**-0.684**	**0.494000**
Son4	**0.07866**	**0.03799**	**2.071**	**0.038859 * **
Height	**0.07692**	**0.04445**	**1.730**	**0.084143 . **
Body.mass.index	**0.08767**	**0.04311**	**2.034**	**0.042490 * **

Signif. codes: 0 '*' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1**

The reason of absence from 1 to 28 except 2 and 16 should be tackled or decreased to reduce absenteeism time in hours. The transportation expenses should also be decreased to reduce Absenteeism time in hours.

If the employee has only one son then absenteeism time in hours decreases which is good for the company. so company can use this as a reference to hire employees whereas if the employee has 4 sons then absenteeism time in hours increases Another parameter is the body mass index which shows that if it increases the absenteeism time in hours increases so company can encourage employee to be physically fit as BMI is an indication of ratio to the height and weight.

3.1.1 Root Mean Squared Error (RMSE)

As to answer the second question put forth by the company it has asked to predict the loss for the company every month if the same trend continues of employee absenteeism. The mean square error which shows us how the predicted values deviate from the actual values will help us to answer this question. The decision tree for prediction model gives the RMSE as 0.1661673 and from the regression model it is 0.1835 the better one would be to select the Decision tree regression model as it is giving lesser value. The calculation is given in the appendix part c which gives us the predicted value of loss per month which is 140.73 hours per month**.** It is worth noting that RMSE is helpful specially while predicting the Time series variant data.

3.2 Model Selection

We can see that linear regression model helps to answer the first question while the decision tree for regression helps to answer the second question asked by the company. So we need to select both the model for our analysis.

Appendix A - R Code and Python Code

```
rm(list = ls(all = T))
getwd()

rm(list=ls())

#Set the directory
setwd("C:/Users/Asus/Documents/R programming")
getwd()

#Load libraries
#Load Libraries
x = c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "C50", "dummies", "e1071", "Information",
    "MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine', 'inTrees')


#install.packages(x)
lapply(x, require, character.only = TRUE)
rm(x)

##Read the data
data_absent = read.csv("Absenteeism_at_work_Project.csv", header = T, na.strings = c(" ", "", "NA"))

#########Converting data types ##############
str(data_absent) # Checking the required data types

data_absent$Reason.for.absence = as.factor(data_absent$Reason.for.absence)
data_absent$Month.of.absence = as.factor(data_absent$Month.of.absence)
data_absent$Day.of.the.week = as.factor(data_absent$Day.of.the.week)
data_absent$Seasons = as.factor(data_absent$Seasons)
data_absent$Disciplinary.failure = as.factor(data_absent$Disciplinary.failure)
data_absent$Education = as.factor(data_absent$Education)
data_absent$Son = as.factor(data_absent$Son)
data_absent$Social.drinker = as.factor(data_absent$Social.drinker)
data_absent$Social.smoker = as.factor(data_absent$Social.smoker)
data_absent$Pet = as.factor(data_absent$Pet)

str(data_absent) # checking the required data types again

##################Missing values analysis#################################
#Create a dataframe with missing percentage
missing_val = data.frame(apply(data_absent,2,function(x){sum(is.na(x))}))

#Convert row names into columns
missing_val$Columns = row.names(missing_val)
```

```
row.names(missing_val) = NULL

#Rename the variable name
names(missing_val)[1] =  "Missing_percentage"

#Calculate the percentage
missing_val$Missing_percentage = (missing_val$Missing_percentage/nrow(data_absent)) * 100

#Arrange in descending order
missing_val = missing_val[order(-missing_val$Missing_percentage),]

#Rearrange the column names
missing_val = missing_val[,c(2,1)]

#plot bar grapghfor missing values
ggplot(data = missing_val[1:3,], aes(x=reorder(Columns, -Missing_percentage),y = Missing_percentage))+
  geom_bar(stat = "identity",fill = "grey")+xlab("Parameter")+
  ggtitle("Missing data percentage (Absent)") + theme_bw()


#Actual value = 0
#predicted value = 31


##create missing value
data_absent$Disciplinary.failure[169]

data_absent$Disciplinary.failure[169] = NaN

################KNN method
data_absent = knnImputation(data_absent, k = 7)

data_absent$Disciplinary.failure[169]


########################################Outlier
Analysis#############################
# ## BoxPlots - Distribution and Outlier Check
numeric_index_absent = sapply(data_absent,is.numeric) #selecting only numeric

numeric_data_absent = data_absent[,numeric_index_absent]

cnames_absent = colnames(numeric_data_absent)

##cnames_direct                                                                    =
c("Transportation.expense","Distance.from.Residence.to.Work","Service.time","Age","Work.loa
d.Average.day","Weight","Height","Body.mass.index")

############## Outlier analysis ################
```

```
for (i in 1:length(cnames_absent))
{
  assign(paste0("gn",i), ggplot(aes_string(y = (cnames_absent[i])), data = subset(data_absent))+
         stat_boxplot(geom = "errorbar", width = 0.5) +
         geom_boxplot(outlier.colour="red", fill = "grey" ,outlier.shape=18,
                      outlier.size=1, notch=FALSE) +
         theme(legend.position="bottom")+
         labs(y=cnames_absent[i])+
         ggtitle(paste("Box plot of Absenteeism for",cnames_absent[i])))
}


## Plotting plots together
gridExtra::grid.arrange(gn1,ncol=1)
gridExtra::grid.arrange(gn2,ncol=1)
gridExtra::grid.arrange(gn3,ncol=1)
gridExtra::grid.arrange(gn4,ncol=1)
gridExtra::grid.arrange(gn5,ncol=1)
gridExtra::grid.arrange(gn6,ncol=1)
gridExtra::grid.arrange(gn7,ncol=1)
gridExtra::grid.arrange(gn8,ncol=1)
gridExtra::grid.arrange(gn9,ncol=1)
gridExtra::grid.arrange(gn10,ncol=1)



###Creating boxplot of each variable #############
boxplot(data_absent$ID)
boxplot(data_absent$Transportation.expense)
boxplot(data_absent$Distance.from.Residence.to.Work)
boxplot(data_absent$Service.time)
boxplot(data_absent$Age)
boxplot(data_absent$Work.load.Average.day)
boxplot(data_absent$Hit.target)
boxplot(data_absent$Disciplinary.failure)
boxplot(data_absent$Education)
boxplot(data_absent$Son)
boxplot(data_absent$Social.drinker)
boxplot(data_absent$Social.smoker)
boxplot(data_absent$Pet)
boxplot(data_absent$Weight)
boxplot(data_absent$Height)
boxplot(data_absent$Body.mass.index)
boxplot(data_absent$Absenteeism.time.in.hours)



############## loop to remove outlier and impute using Knn#############
for(i in cnames_absent)
  {val = data_absent[,i][data_absent[,i] %in% boxplot.stats(data_absent[,i])$out]
```

```
  #print(length(val))
  data_absent[,i][data_absent[,i] %in% val] = NA
  }

data_absent = knnImputation(data_absent, k = 7)

###################################Feature Selection##############################
## Correlation Plot
corrgram(data_absent[,numeric_index_absent], order = F,
        upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")

####Drop Body mass index ######

## Chi-squared Test of Independence
factor_index = sapply(data_absent,is.factor)
factor_data = data_absent[,factor_index]

for (i in 1:11)
{
  print(names(factor_data)[i])
  print(chisq.test(table(data_absent$Absenteeism.time.in.hours ,factor_data[,i])))
}

## Dimension Reduction
data_absent = subset(data_absent,
                     select = -
c(ID,Weight,Day.of.the.week,Education,Social.smoker,Social.drinker,Pet,Work.load.Average.da
y))

str(data_absent)

###################################Feature Scaling################################
#Normality check
qqnorm(data_absent$Transportation.expense)
hist(data_absent$Transportation.expense) #data is right skewed

numeric_index_norm = sapply(data_absent,is.numeric) #selecting only numeric

numeric_data_norm = data_absent[,numeric_index_norm]

cnames_absent_norm = colnames(numeric_data_norm)

cnames_norm =
c("Transportation.expense","Distance.from.Residence.to.Work","Service.time","Age","Hit.target
","Height","Body.mass.index","Absenteeism.time.in.hours")

for(i in cnames_norm){
  print(i)
  data_absent[,i] = (data_absent[,i] - min(data_absent[,i]))/
    (max(data_absent[,i] - min(data_absent[,i])))
```

```
}

df = data_absent
#Write output results back into disk
write.csv(data_absent, "data_absent_new.csv", row.names = F)


#####################################Train and test data ##################
###############Clean the environment
library(DataCombine)
rmExcept("df")

#Divide data into train and test using stratified sampling method
set.seed(1234)
train.index = createDataPartition(df$Absenteeism.time.in.hours, p = .80, list = FALSE)
train = df[ train.index,]
test  = df[-train.index,]

#Load Libraries
library(rpart)
library(MASS)

#Linear regression
#check Multicollinearity

library(usdm)
vif(df[, -13])
numeric_index = sapply(df,is.numeric) #selecting only numeric

numeric_data = df[,numeric_index]

cnames_df = colnames(numeric_data)
cnames_df = data.frame(cnames_df)

vifcor(cnames_df_numeric[, -8], th = 0.9)

#Build regression model on train data
lm_model = lm(Absenteeism.time.in.hours ~., data = train)

#Summary of the model
summary(lm_model)

#Predict the values of test data by applying the model on test data
predictions_LR = predict(lm_model , test[,1:9])

#Calculate MAPE ( mean absolute percentage error)
mape(test[,10], predictions_LR)

####### ##rpart for regression
fit = rpart(Absenteeism.time.in.hours ~ ., data = train, method = "anova")
```

```
#Predict for new test cases
predictions_DT = predict(fit, test[,-13])

#Alternate method
regr.eval(test[,'Absenteeism.time.in.hours'] , predictions_DT ,stats = c('mae','rmse','mape','mse'))

# ####################Model Performance ####################
fit
plot(fit)
varImp(fit)*100
pred <- predict(fit, newdata = test)
RMSE(predictions_DT, test$Absenteeism.time.in.hours)
plot(predictions_DT ~ test$Absenteeism.time.in.hours)

####################################Decision Tree Regression ####################

#########rpart for regression ####################
fit_dt = rpart(Absenteeism.time.in.hours~ ., data = df, method = "anova")

#Predict for new test cases
#Predict for new test cases
predictions_DT = predict(fit_dt, test[,-13])

RMSE(predictions_DT, test$Absenteeism.time.in.hours)
```

Python code:

```
 #Load libraries
import os
import pandas as pd
import numpy as np
from fancyimpute import KNN
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency
import seaborn as sns
#set working directory
os.chdir("F:\Others\Project Absenteesim")
#Load data
data_absent = pd.read_csv("Absenteeism_at_work_project.csv")
data_absent.dtypes
#Create dataframe with missing percentage
missing_val = pd.DataFrame(data_absent.isnull().sum())
#Reset Index
missing_val = missing_val.reset_index()
#Rename variables
missing_val = missing_val.rename(columns={'index':'Variables', 0 : 'Missing_Percentage'})
#calculate percentage
missing_val["Missing_Percentage"] = (missing_val['Missing_Percentage']/len(data_absent))*100
#desecnding order
missing_val = missing_val.sort_values('Missing_Percentage', ascending = False).reset_index(drop = True)
data_absent.head(10)
data_absent["Disciplinary failure"].loc[70]
data_absent["Disciplinary failure"].loc[70] = np.nan
#KNN Imputation
#Assigining level to categories
for i in range(0, data_absent.shape[1]):
   #print(i)
   if(data_absent.iloc[:,i].dtypes == 'object'):
```

```
    data_absent.iloc[:,i] = pd.Categorical(data_absent.iloc[:,i])
    #Print(marketing_train[[i]])
    data_absent.iloc[:,i] = data_absent.iloc[:,i].cat.codes
```

```
data_absent.head(10)
#KNN imputation
data_absent = pd.DataFrame(KNN(k = 7).complete(data_absent), columns = data_absent.columns)
data_absent.head(10)
data_absent["Disciplinary failure"].loc[70]
#plot boxplot to visualize outliers
%matplotlib inline


plt.boxplot(data_absent['Transportation expense'])
#Detect and replace with NA
#Extract quartiles
q75, q25 = np.percentile(data_absent['Transportation expense'], [75 ,25])
```

```
#Calculate Iqr interquartile range
iqr = q75 - q25
```

In [ ]:

```
#calculate inner and outer fence
min = q25 - (iqr*1.5)
max = q75 + (iqr*1.5)
#Replace values which are above and below the fence with NA
data_absent.loc[data_absent['Transportation expense'] < min,:'Transportation expense'] = np.nan
data_absent.loc[data_absent['Transportation expense'] > max,:'Transportation expense'] = np.nan
#Calculate missing value
missing_val_absent = pd.DataFrame(data_absent.isnull().sum())
```

```
#Impute with KNN
data_absent = pd.DataFrame(KNN(k = 7).complete(data_absent), columns = data_absent.columns)
data_absent.isnull().sum()
```

```
data_absent.head(5)
#Save numeric variables
cnames = ["ID","Transportation Expense", "Distance from residence to work","Service time","Age","Weight","Height","Body mass index","Absenteeism time in hours"]
##Correlation plot
#Corelation plot
df_corr = data_absent.loc[:,cnames]
#Set the width and height of plot
f , ax = plt.subplots(figsize =(7,5))

#Set correlation matrix
corr = df_corr.corr()

#Plot using seaborn library
sns.heatmap(corr, mask = np.zeros_like(corr ,dtype = np.bool), cmap=sns.diverging_palette(220, 10, as_cmap=True),
        square=True, ax=ax)
data_absent.dtypes
##Chi square test
#Save categorical variable
cat_names = ["Reason for absence","Month of absence","Day of the week","Seasons","Work load Average/day ","Hit target","Disciplinary failure","Education","Son","Social drinker","Social smoker","Pet"]
cat_names
#loop for chi square values
for i in cat_names:
   print(i)
   chi2, p, dof, ex = chi2_contingency(pd.crosstab(data_absent['Absenteeism time in hours'],data_absent[i]))
   print(p)
#Drop the variables from data
data_absent = data_absent.drop(['ID','Weight','Day of the week','Education','Social smoker','Social drinker','Pet'], axis=1)
data_absent = data_absent.drop(['Work load Average/day'])
data_absent.head(3)
del data_absent['Work load Average/day']
#Normalisation
```

```
for i in cnames:
   print(i)
   data_absent[i] = (data_absent[i] - min(data_absent[i]))/(max(data_absent[i]) - min(data_absent[i]))
##############decision tree for regression############


#Divide data into train and test
train , test = train_test_split(data_absent , test_size = 0.2)
#decision tree for regresion
fit_DT = DecisionTreeRegressor(max_depth = 2).fit(train.iloc[:,0:9], train.iloc[:,9])
#Apply model on test data
predictions_DT = fit_DT.predict(test.iloc[:,0:9])
#Calculate MAPE
def MAPE(y_true, y_pred):
   mape = np.mean(np.abs((y_true - y_pred)/y_true))*100
   return mape
MAPE(test.iloc[:,9], predictions_DT)
######################### Regression Model ################
#Import libraries for LR
import statsmodels.api as sm

#Train the model using the training sets
model = sm.OLS(train.iloc[:,9],
           train.iloc[:,0:9]).fit()
#Print out the summary
model.summary()
#make the predictions by the model
predictions_LR = model.predict(test.iloc[:,0:9])#Calculate MAPE
MAPE(test.iloc[:,9],                                                    predictions_LR)
```

Appendix B: Calculation for total loss per month

Actual test data Value sum = sum(test$Absenteeism.time.in.hours) = 40.56

Predicted test data Value sum = sum(PredicDTt$PredictDT) = 41.72

Observation no 4 of absenteeism time in hours before scaling = 2 hours

Observation no 4 of absenteeism time in hours after scaling = 0.25 hours

Factor 1 = 2/0.25 = 8

Total no of observation = 740

Predicted no of observation = 146

Factor 2= 740/146 = 5.06

sum of predicted values of Absenteeism time in hours for 146 observation = 41.72

hence , sum for 740 observation = 41.72 * 5.06 = 211.10

sum of absenteeism time in hours without scaling = 211.10 * factor 1 = 211.10*8 = 1688.8256

Total hours of absenteeism per month = 1688.8256/12 = 140.73

References

1. https://stackoverflow.com/users/10254950/sarang-kapse

2. https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd

3. https://www.rdocumentation.org/packages/Metrics/versions/0.1.4/topics/rmse

4. https://www.youtube.com/watch?v=tSPg-JDAF4M

5. http://www.sthda.com/english/articles/40-regression-analysis/167-simple-linear-regression-in-r/